

I Claim:

1. A method for altering the execution flow of a computer program comprising the steps of:
 - 5 establishing a breakpoint in the computer program;
 - initiating execution of the computer program;
 - 10 interrupting execution of the computer program at an origin address where the breakpoint is detected; and
 - changing the instruction pointer to point to a destination address specified by the breakpoint; and
 - 15 continuing execution of the computer program from the destination address.
2. The method as claimed in claim 1, further comprising the step of storing an
20 original instruction from the origin address.
3. The method as claimed in claim 2, further comprising the step of replacing the original instruction at the origin address with a break instruction.
- 25 4. The method as claimed in claim 3, further comprising the step of restoring the stored original instruction at the origin address after the step of interrupting execution of the computer program and before the step of changing the instruction pointer.
- 30 5. The method as claimed in claim 4, further comprising the step of repeating the step of replacing the original instruction at the origin address with the break instruction, after the step of changing the instruction pointer.

6. The method as claimed in claim 4, wherein the execution of the computer program continues in a single-stepping mode before the step of changing the instruction pointer.
- 5 7. The method as claimed in claim 1, further comprising the step of adding the breakpoint address to a debugging register.
8. The method as claimed in claim 7, further comprising the step of removing the breakpoint address from a debugging register.
- 10 9. The method as claimed in claim 1, further comprising the step of clearing the breakpoint from the computer program.
- 15 10. The method as claimed in claim 1, wherein the breakpoint further specifies the origin address.
11. The method as claimed in claim 1, wherein the breakpoint further specifies the memory address that stores the original instruction.
- 20 12. The method as claimed in claim 1, further comprising the step of loading the computer program as a child process of a debugger application.
13. The method as claimed in claim 1, further comprising the step of determining whether the breakpoint specifies a destination address that diverts execution flow of the computer program.
- 25 14. The method as claimed in claim 1, wherein the destination address specified by the breakpoint is provided from user input.
- 30 15. A computer program product, recorded on a medium, for altering the execution flow of a computer program, the computer program product comprising software for performing the steps of:

establishing a breakpoint in the computer program;

initiating execution of the computer program;

5 interrupting execution of the computer program at an origin address
where the breakpoint is detected; and

 changing the instruction pointer to point to a destination address
specified by the breakpoint; and

10

 continuing execution of the computer program from the destination
address.

16. A computer system for altering the execution flow of a computer program, the
15 computer system executing computer software for performing the steps of:

 establishing a breakpoint in the computer program;

 initiating execution of the computer program;

20

 interrupting execution of the computer program at an origin address
where the breakpoint is detected; and

 changing the instruction pointer to point to a destination address
25 specified by the breakpoint; and

25

 continuing execution of the computer program from the destination
address.